REMARKS/ARGUMENTS

Claims 1-46 stand rejected under 35 USC 102(e) as being anticipated by Tanaka et al (US Patent No. 6,633,538). The rejection under 35 USC 102(e) is respectfully traversed for the reason that each of independent claims 1, 18, 25, 30, 33, 34, 36, 40 and 41 contain at least one limitation not taught in Tanaka et al.

Claims 1, 18, 25, 30, 33, 34, 36, 40 and 41 all include the following limitations: a plurality of high-availability aware components, registration of the high-availability aware components, and dynamic allocation of roles and assignments to one or more of the registered high-availability aware components.

A high-availability aware component is defined in the specification at page 12, paragraph [0028]:

> A component is an encapsulation of a logical aggregation of functions provided by software, hardware, or both that is designated to behave as a unit of deployment, redundancy, and manageability within a networked computer system. A component may have various states ("component states"). It may be uniquely identified by its type ("component type"), representing the functional semantic of the component. A component type may be instantiated into one or multiple component instances. A component instance, which may be referred to as a component for simplicity, acts as the unit of manageability of the CRIM.

Further description of the high-availability aware components are found, inter alia, in paragraphs [0029] through [0054].

It is clear from a fair reading of paragraphs [0028] through [0054] that a "high-availability aware component" as claimed is a sub-component of a computer

system and not an actual node as taught in Tanaka et al. As is known to persons of ordinary skill in the art a "node" is generally defined as a computer system in a network of computer systems. Each node usually consists of memory, a processing unit, I/O devices, and is meant to communicate with the other nodes over the network, via specific network addressing and protocol.

Tanaka's teaching considers a node, a computer system with one or more IP addresses, as the "black-box" unit of availability management. This is in marked contrast to the present invention, wherein a "high-availability aware component" is not a node, but a sub-component of a node. Therefore, in the present invention, availability management is accomplished at a much finer granularity, so that some components in a node can be in a primary state while other components on the same node can be in a secondary or spare state.

Registration of the high-availability aware components is described in the present specification, inter alia, at paragraphs [0065], [0073], and Table 3. It is clear from a fair reading of these paragraphs that when a component registers with the availability manager, it not only announces its readiness to accept work, but also provides the means (more specifically callbacks) through which the availability manager can communicate with the component in order to manage the component's state and as well as its assignments (i.e., its workload). (See Paragraph [0036] for the definition of "component assignment".) The registration of a component here is an action (execution of interface software) on a software entity - the availability manager.

In contrast, Tanaka et al teaches, at column 5, only that the interaction between the master node and other nodes are at the node (IP address) level and the master node gives the node its state and the resources encapsulated in a node (which must identify its workload) are identified in the configuration table (denoted as a schedule table in Tanaka). Tanaka teaches "configuration information", especially the IP addresses representing the nodes. See columns 5 and 6 of

Tanaka. Tanaka does not teach "registration" as claimed. Applicants have carefully read column 5 of Tanaka and can find no teachings of "registration" as claimed.

Finally, dynamic allocation of roles and assignments is described in the present specification, inter alia, at paragraphs [0055], [0056], [0075], [0103], [0104], and [0110].

It is clear from a reading of these paragraphs that allocation is defined as assigning a computer the following:

(i) its role such as primary, secondary, spare,

(ii) its assignment (in generic term its workload). Note that the availability management dynamically decides on the assignments given to a component, and it may assign one or more assignments dynamically. In other words, associating "component assignments" (i.e., workloads) to a computer is accomplished dynamically and its targets are the plurality of components on a computer.

The present invention differs from Tanaka's teaching by:

(i) component-level allocation (compared to a node level) and

(ii) requiring dynamic allocation of "component assignments" to components.

It is not well understood how Tanaka et al is able to dynamically allocate roles and assignments to the components, since Tanaka et al does not teach "high availability aware components" as claimed. In contrast, Tanaka teaches, at column 6, only that the functions of a node is defined in the configuration and does not change dynamically and is not "allocated" as claimed. Applicants have carefully read column 6 of Tanaka and can find no teachings of "allocation" as claimed.

In view of all of the above, the claims are now believed to be allowable and the case in condition for allowance which action is respectfully requested. Should the Examiner be of the opinion that a telephone conference would expedite the

prosecution of this case, the Examiner is requested to contact Applicants' attorney at the telephone number listed below.

No fee is believed due for this submittal. However, any fee deficiency associated with this submittal may be charged to Deposit Account No. 50-1123.

Respectfully submitted,

1/28, 2005

Peter J. Meza, No. 32,920
Hogan & Hartson LLP
One Tabor Center
1200 17th Street, Suite 1500
Denver, Colorado 80202
(719) 448-5906 Tel
(303) 899-7333 Fax